Mining Frequent Items Using Directed Graphs

B.Bhargavi, B.Venkanna, and V.Hari Prasad

Abstract — On the basis of the association rule mining and Apriori algorithm, this paper proposes an improved algorithm. Apriori Algorithm is used to mine association rules. Apriori is designed to operate on databases containing transactions. As is common in association rule mining, given a set of itemsets, the algorithm attempts to find subsets which are common to at least a minimum number of the itemsets. The improved algorithm promotes the efficiency of the algorithm by reduced scanning of the datasets and improving the efficiency of the pruning step. Based on the concept of strong rules, Agrawal introduced association rules for discovering regularities between products in large scale transaction data recorded by point-of-sale (POS) systems in supermarkets. Such information can be used as the basis for decisions about marketing activities such as promotional pricing or product placements.

Keyword:- **association rule; Apriori algorithm; Directed Graph**

I. INTRODUCTION

Mining Association rule is very important field of research in data mining. Data mining is a technology of multi-interdisciplinary research field, which combines the latest research results in database technology, artificial intelligence, machine learning, statistics, knowledge engineering, information retrieval, high-performance computing, and data visualization technology and so on. In computer science and data mining, Apriori is a classic algorithm for learning association rules. Apriori is designed to operate on databases containing transactions. As is common in association rule mining, given a set of itemsets, the algorithm attempts to find subsets which are common to at least a minimum number C of the itemsets. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. The problem of mining Association rule is put forward by R.S Agarwal first in 1993. Now the Association rules are widely applied in E-commerce, bank credit, shopping cart analysis, market analysis, fraud detection, and customer retention, to production control and science exploration, etc., Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. It generates candidate item sets of length from item sets of length. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent -length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

II. ASSOCIATION RULES MINING

Association rule finds interesting associations and/or correlation relationships among large set of data items. Association rule shows attribute value conditions that occur frequently together in a given dataset. A typical and widely-used example of association rule mining is Market Basket Analysis. Association rules are the rules that describe potential relationship of data items in the database. The discovery of association rules is the most common task in data mining.

A. The Definition of the Association Rules Definition 1: Let I= {i1，i2，…，im} be a set of items.
Let D the task-relevant data, be a set of database transactions where each transaction T is a set of items such that T⊆I. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if A⊆T.
Definition 2: An association rule is an implication of the form A=>B, where A⊆I, B⊆I and A∩B=Φ. The rule A=>B holds in the transaction set D with support s, where s is the percentage of transactions in D that contain A⊆B.
This is taken to be the probability, P (A⊆B). The rule A=>B has confidence c in the transaction set D, where c is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability, P (A|B). That is
Support (A=>B) = P (A⊆B)
Confidence (A=>B) = P (A|B)
Definition 3: Rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (*min_conf*) are called strong.

B. Basic Concept of Association Rules

Following the original definition by Agrawal the problem of association rule mining is defined as: Let I = {i1, i 2, ..., i n} be a set of n binary attributes called items . Let D = {t1, t2, ..., tn} be a set of transactions called the database . Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form X→Y where X, Y□ I and X∩Y = □. The sets of items (for short itemsets) X and Y are called antecedent (left-hand-side or LHS) and consequent (right-hand-side or RHS) of the rule respectively. To illustrate the concepts, we use a small example from the supermarket domain. The set of items is I = {milk,bread,butter,beer} and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table below. An example rule for the supermarket could be {milk,bread}=>{butter} meaning that if milk and bread is bought, customers also buy butter. Note: this example is extremely small. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

Transaction ID	Milk	Bread	Butter	Beer
1	1	1	0	0
2	0	1	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

B.Bhargavi is a PG student, B.Venkanna, Faculty, is working as Assistant Prof and V.Hari Prasad, HOD of CSIT, working as Associate Prof, Authore are from Spoorthy College of Engg & Tech, Hyderabad,Email id: boppidi.bhargavi@gmail.com

6	1	0	0	0
7	0	1	1	1

Useful Concepts

To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best-known constraints are minimum thresholds on support and confidence. Support The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the itemset.

supp(X)= no. of transactions which contain the itemset X / total no. of transactions

In the example database, the itemset {milk,bread,butter} has a support of 4 /15 = 0.26 since it occurs in 26% of all transactions. To be even more explicit we can point out that 4 is the number of transactions from the database which contain the itemset {milk,bread,butter} while 15 represents the total number of transactions.

Confidence: The confidence of a rule is defined:

Conf(x->y) = supp(x U y)/supp(x)

For the rule {milk,bread}=>{butter} we have the following confidence:

supp({milk,bread,butter}) / supp({milk,bread}) = 0.26 / 0.4 = 0.65

This means that for 65% of the transactions containing milk and bread the rule is correct. Confidence can be interpreted as an estimate of the probability P(Y | X), the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

Lift The lift of a rule is defined as:

Lift(x->y) = supp(x U y)

Supp(x) * sup(y)

The rule {milk,bread}=>{butter} has the following lift: supp({milk,bread,butter}) / supp({butter}) x supp({milk,bread})= 0.26/0.46 x 0.4= 1.4.

Conviction The conviction of a rule is defined as:

Conv(x->y) = 1 – supp(y)
----
1 – conf(x->y)

The rule {milk,bread}=>{butter} has the following conviction: 1 – supp({butter})/ 1- conf({milk,bread}=>{butter}) = 1-0.46/1-0.65 = 1.54 The conviction of the rule X=>Y can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions. In this example, the conviction value of 1.54 shows that the rule {milk,bread}=>{butter} would be incorrect 54% more often (1.54 times as often) if the association between X and Y was purely random chance.

C. Apiori Algorithm:

Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent itemsets in a database.
2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

Mining association rules is to find out strong association rules that satisfy minimum support and minimum confidence in the transaction database D. It can be decomposed into two steps. First, find out all the frequent itemsets in transaction database D. Second, generate strong association rules from the frequent itemsets. The key to the discovery of strong association rules is finding out frequent itemsets, while Apriori algorithm is the most classical algorithm for the search of frequent itemsets.

Apriori employs an iterative approach known as a level wise search, where k-itemsets are used to explore (k+1)- itemsets. First, the set of frequent 1-itemset L1 is found. Next, L1 is used to find frequent 2-itemset L2. Then L2 is used to find frequent 3-itemset L3. Iterate like this until no more frequent k-itemset can be found. The finding of Lk requires a full scan of the database. To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property, presented is used to reduce the search space. Apriori property: All nonempty subsets of a frequent itemset must also be frequent. A two-step process is used to find the frequent itemsets: join and prune actions.

(1) The join step

To find Lk, a candidate k-itemset Ck can be generated by joining Lk-1 with itself.

(2) The prune step

If a (k-1)-dimension subset of X in a candidate itemset Ck is not frequent, then according to the property of Apriori algorithm, we know that X is not frequent either and should be deleted from Ck.

The Example Illustrates the Traditional Apriori Algorithms Transaction database as shown in Table I, minsup =50%, minconf =70%.Request the frequent association rules in transaction database D.

TABLE I. THE TABLE OF TRANSACTION DATABASE

TID	ITEMSETS
1	A B CD E
2	A B C
3	CD E F
4	A B E

The implementation process is as follows:

Step 1: Find the frequent itemsets

L=Ll U L2 U L3={ {A }, {B }, {C },{D}, {E },{AB }, {AC}, {AE}, {BC}, {BD}, {CD}, {CE }, {ABC} }.

Step 2: Get association from {ABC} Only

{AC} ⊆{B}, {BC} ⊆ {A}, {A} ⊆{B}, {B} ⊆ {A} meet the requirements. Confidence level is 100%. Time Complexity Analysis: Scan database 3 times; Data Processing of database 15 times to get frequent l-itemset; Data Processing 19 times to get frequent 2-itemset; Data Processing of database 11 times to get frequent 3-itemset, that is 45 times data processing in all.

III. ALGORITHM BASED ON THE DIRECTED GRAPH

Directed Graph:

Graph is a sort of complicated non linear structure. Graphs are used to solve problems in many fields, such as artificial intelligence, mathematics, physics, chemistry, biology, and computer science. A directed graph G = (V, E) consists of a finite set of V (V≠Φ) of vertices and a finite set of E of edges that are ordered pairs of elements of V, denoted as V (G) and E (G). Each directed edges e= <v_i, v_j > of G has a number w_{ij} (w_{ij}>=0) is called directed network (or weighted graphs). The distance matrix $D_{n\times n}$= [d_{ij}] of the directed graph G is defined as follows:

$$d_{ij} = \begin{cases} w_{ij} & \exists directed \quad edge, e = \langle v_i, v_j \rangle \in E \\ \infty & else \end{cases}$$

The Path is the set of paths in the directed Graph G = (V, E), a path is defined as a road from v_i to v_j in a graph with directed edges.

B. The improved algorithm

The improved algorithm only scans the dataset D once, find the directed graph G, where the set of vertices V register the itemset, namely V (G) =I; the set of edges E register the edges which are composed by the pair of items in a transaction, E(G)= <I_i, I_j>; the weight of each edge is the support count of each edge which get when scan the dataset D; the set of paths P register the longest path in each transaction and its support count. We get the frequent itemsets by the distance matrix and path searching. By convention, the improved algorithm assumes that items within a transaction or itemset are sorted in lexicographic order. Enter: dataset D, itemset I and the minimum threshold of support Min-sup. Output: all the frequent itemsets L.

```
Method:
(1) Scan the dataset D; // without frequent 1-itemset
(2) G=(V,E), Dn×n=[dij]，Path
(3) for each dij {
(4) if wij>=Min_sup then add dij to L2
(5)}
(6) for (i=3; |Li-1|>1; i++) {//more than one itemset in Li-1
(7) Li= gen_freq (Path, Li-1, Min_sup);
//get the frequent itemsets
(8)}
(9) return L=Ui Li;
Procedure gen_freq (Path, Li-1, Min_sup)
// for join and prune
(1) for each itemset l1ϵLi-1
(2) for each itemset l2ϵLi-1
(3) if l1∞l2 then{
(4) l= (l1[1], l1[2],…, l1[i-1], l2[i-1]); //path
(5) if judge_frequent  (l, Path) then
(6) add l to Li;
(7) else delete l; //prune
(8)}
(9) return Li;

Procedure judge_frequent (l, Path)
(1) if (lϵPath) ^ (wl>=Min_sup) then
//find the path l and its support count
(2) return true;
(3) else return false;
```

IV. SAMPLE USAGE OF APIORI ALGORITHM

A large supermarket tracks sales data by Stock-keeping unit (SKU) for each item, and thus is able to know what items are typically purchased together. Apriori is a moderately efficient way to build a list of frequent purchased item pairs from this data. Let the database of transactions consist of the sets {1,2,3,4}, {1,2,3,4,5}, {2,3,4}, {2,3,5}, {1,2,4}, {1,3,4}, {2,3,4,5}, {1,3,4,5}, {3,4,5}, {1,2,3,5}. Each number corresponds to a product such as "butter" or "water". The first step of Apriori is to count up the frequencies, called the supports, of each member item separately:

Item	Support
1	6
2	7
3	9
4	8
5	6

We can define a minimum support level to qualify as "frequent," which depends on the context. For this case, let min support = 4. Therefore, all are frequent. The next step is to generate a list of all 2-pairs of the frequent items. Had any of the above items not been frequent, they wouldn't have been included as a possible member of possible 2-item pairs. In this way, Apriori prunes the tree of all possible sets. In next step we again select only these items (now 2-pairs are items) which are frequent (the pairs written in bold text):

Item	Support
{1,2}	4
{1,3}	5
{1,4}	5
{1,5}	3
{2,3}	6
{2,4}	5
{2,5}	4
{3,4}	7
{3,5}	6
{4,5}	4

We generate the list of all 3-triples of the frequent items (by connecting frequent pair with frequent single item).

Item	Support
{1,3,4}	4
{2,3,4}	4
{2,3,5}	4
{3,4,5}	4

The algorithm will end here because the pair {2,3,4,5} generated at the next step does not have the desired support. We will now apply the same algorithm on the same set of data considering that the min support is 5. We get the following results:

Step 1:

Item	Support
1	6
2	7
3	9
4	8
5	6

Step 2:

Item	Support
{1,2}	4
{1,3}	5
{1,4}	5
{1,5}	3
{2,3}	6
{2,4}	5
{2,5}	4
{3,4}	7
{3,5}	6
{4,5}	4

The algorithm ends here because none of the 3-triples generated at Step 3 have de desired support.

V. APPLICATIONS

APPLICATIONS OF DATAMINING:

Data mining have been applied in various research works. One of the popular techniques used for mining data in KDD for pattern discovery is the association rule [1]. According to [2] an association rule implies certain association relationships among a set of objects. It attracted a lot of attention in current data mining research due to it's capability of discovering useful patterns for decision support, selective marketing, financial forecast, medical diagnosis and many other application. The association rules technique works by finding all rules in a database that satisfies the determined minimum support and minimum confidence [3].A Software training center applies Apriori algorithm to find the frequently opted courses by the students. This paper will discuss the results of a pattern extraction process using association rules of data mining technique using Apriori Algorithm

PATTERN EXTRACTION:

In this study the KDD process which involved several steps: Selection, Pre-processing, Transformation, Pattern Extraction using data mining, and Interpretation/Evaluation [8].

A) Selection

These data have been generated by a software training center. The initial data contains Student_id, Name, course_id, course_name. The data contains various types of values either string or numeric value. The selected attributes are Student_id and course_name. The data were then processed for generating rules.

B) Pre-processing

Upon initial examination on the data, missing values of the attributes *Student_id and course_name* were found and removed according to the numbers of missing values in one instance as part of the data cleansing process.

C) Transformation

After the cleansing process, data is converted into a common format to make sure that the data mining process can be easily performed besides ensuring a meaningful result produced. Rules are applied to transform the *course_name* to uniform string data. For example:

If the course_name=c++ or Tourbo c++ THEN Replace *course_name* by CPP

D) Pattern Extraction using Apriori Algorithm

In this study, the association rules using Apriori Algorithm was applied to the data for generating rules using WEKA software.

E) Interpretation/ Evaluation

During the process of pattern extraction, the acceptance of the output produced was evaluated in terms of accuracy and converge. This is to make sure that the generated rules are reliable and accurate. The accuracy of rules was obtained according to the value of confidence parameter determined earlier in the study while the degree of rules coverage was shown through the value of support parameter

Table 4 shows pre-processed data of a training center after applying the improved algorithm the result we achieve is {.Net, Sql} by applying the minimum support as 2.

Table II: Software Training center data

Student_id	Course_name
100	.Net, Sql
101	.Net, Sql, Jscript
102	DreamWeaver, Flash, Jscript
103	.Net, Sql
104	CPP, Java

APPLICATIONS OF APRIORI ALGORITHM:

1. Application for drug reaction detection:

The objective is to use the Apriori association analysis algorithm for the detection of adverse drug reactions (ADR) in health care data. The Apriori algorithm is used to perform association analysis on the characteristics of patients, the drugs they are taking, their primary diagnosis, co-morbid conditions, and the ADRs or adverse events (AE) they experience. This analysis produces association rules that indicate what combinations of medications and patient characteristics lead to ADRs.

2. Application in Oracle Bone Inscription Explication:

Oracle Bone Inscription (OBI) is one of the oldest writing in the world, but of all 6000 words found till now there are only about 1500 words that can be explicated explicitly. So explication for OBI is a key and open problem in this field. Exploring the correlation between the OBI words by Association Rules algorithm can aid in the research of explication for OBI. Firstly the OBI data extracted from the OBI corpus are pre-processed; with these processed data as input for Apriori algorithm we get the frequent itemset. And combined by the interestingness measurement the strong association rules between OBI words are produced. Experimental results on the OBI corpus demonstrate that this proposed method is feasible and effective in finding semantic correlation for OBI.

VI. CONCLUSION

We proposed a new algorithm by extending the Apriori algorithm based upon Directed graph to mine the association rules from the database. This algorithm shows efficiency and accuracy to mine association information from massive data faster.

In this paper, we mainly used association rules to achieve go and extract the patterns for the last set of data.

VII. REFERENCES

[1] Agrawal, R. and Srikant, R. "Fast Algorithms for Mining Association Rules," Proceeding of 20th International conference on Very Large Database, 1994, pp. 487-499.

[2] Agrawal, R., Imielinski, T., and Swami, A. "Mining Association Rules between Sets of Items in Large Databases," Proceeding of the 1993 ACM SIGMOD International Conference on Management of Data, Vol. 22, Issue 2, June 1993, pp. 207-216.

[3] S. Chai, H. Wang, J. Qiu, "DFR: A New Improved Algorithm for Mining Frequent Item sets", Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)

[4] R. Agrawal, T. Imielinski, A. Swami, "Mining Association Rules between Sets of Items in Very Large Databases [C]", Proceedings of the ACM SIGMOD Conference on Management of Data, Washington, USA, 1993-05: 207-216

[5] R. Agrawal, T. Srikant, "Fast Algorithms for Mining Association Rules in Large Databas [C]", Proceedings of 20th VLDB Conference, Santiago, Chile, 1994: 487-499

[6] L Guan, S Cheng, and R Zhou, "Mining Frequent Patterns without Candidate Generation [C]", Proceedings of SIGMOD'00, Dallas, 2000:1-12